

REMARKS

Reconsideration and allowance of the subject application are respectfully requested.

Upon entry of this Amendment, claims 1-17 are all the claims pending in the application.

Applicant respectfully submits that the claims define patentable subject matter.

I. Overview of the Office Action

Claims 5, 6, 9, and 10 are objected to because of minor informalities. Claims 1-17 are rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. Claims 4, 6, 10, 11, and 12 are rejected under 35 U.S.C. § 102(e) as being anticipated by Baentsch et al (U.S. Patent Application Publication No. 2002/0059475, hereafter "Baentsch"). Claim 7 is rejected under 35 U.S.C. § 102(b) as being anticipated by Kwong et al (U.S. Patent No. 6,289,506). Claims 1-3, 14, 15, and 17 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Zee (U.S. Patent Application Publication No. 2003/0005425) in view of Baentsch. Claims 5 and 13 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Baentsch in view of Kwong. Claims 8, 9, and 16 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Kwong in view of Baentsch.

II. Preliminary Matters

A. Claim Objections

Claims 5, 6, 9, and 10 have been objected to because of alleged informalities. By this Amendment, Applicant has amended claims 5, 6, 9, and 10 in order to add clarity to the claimed invention. Accordingly, the Examiner is respectfully request to remove the claim objections.

B. 35 U.S.C. § 101 rejections

Claims 1-17 have been rejected under 35 U.S.C. § 101 as allegedly being directed to non-statutory subject matter. By this Amendment, Applicant has amended claims 1, 4, 7, 10, and 12 in order to more clearly recite exemplary embodiments of the invention. Support for the claim amendments is found in the specification in at least paragraphs [22] and [104] to [105]. Accordingly, the Examiner is respectfully request to remove the 35 U.S.C. § 101 rejections.

III. Prior Art Rejections

A. Rejection of claims 4, 6, 10, 11, and 12

Claims 4, 6, 10, 11, and 12 are rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Baentsch. Applicant respectfully traverses the rejection.

Claim 4 recites in part:

wherein a symbolic reference information indicates a specific class, field or method of an object, and method information of the method comprises an

attribute of a code formed of the machine instruction having an operand in which the symbolic reference information is inserted in place of an address.

There is no teaching or suggestion in Baentsch of a machine instruction having an operand in which the symbolic reference information is inserted in place of an address as required by claim 4.

Baentsch discloses a Java run-time system comprising a stacked-based interpreter for executing a Java program which comprises Java bytecode instructions and Java class structures (paragraph [0011]). The Java runtime system downloads and links a Java program containing the Java bytecode instructions, Java class structures, and standard Java symbolic linking strings onto linking identifiers to be stored in an expert file, wherein the linking identifiers are suitable to bind references in bytecode instructions to be executed in corresponding link targets (paragraph [0011]).

However, there is no disclosure in Baentsch of a machine instruction having an operand in which the symbolic reference information is inserted in place of an address. The Examiner cites paragraphs [0008]-[0009] of Baentsch as allegedly disclosing this feature of the claim. However, this cited portion of Baentsch is contrary to the claimed invention. Baentsch discloses that during a downloading and linking process, the interpreter of the Java Virtual Machine (JVM) looks up target items referred to in the bytecode instructions by name and **replaces the references** with corresponding addresses (paragraph [0008], lines 11-15), as opposed to the claimed invention which inserts a symbolic reference **in place of** an address. This feature is also analogously recited in claims 10 and 12.

Further, claim 4 is related to Java class file structure including symbolic reference information. However, the cited paragraphs of Baentsch ([0008]-[0009]) disclose functions which the interpreter performs during a downloading and linking process.

Accordingly, Applicant respectfully submits that independent claims 4, 10, and 12 should be allowable, because the cited reference does not teach or suggest all of the features of the claims. Claims 5, 6, 11, and 13 should also be allowable at least by virtue of their dependence on independent claims 4, 10, and 12.

B. Rejection of Claim 7

Claim 7 recites in part:

(a) precompiling a class file included in a standard class library into an extended class library file including a machine instruction; (b) the extended class library file executing the machine instruction; and (c) executing a Java application file by using at least one of a Just-In-Time (JIT) compiling method and an interpreting method.

There is no teaching or suggestion in Kwong of precompiling a class file included in a standard class library into an extended class library file including a machine instruction as required by claim 7.

Kwong discloses a method for optimizing Java performance using precompiled code. The performance of a program code is monitored during program execution (column 2, lines 36-44). A list of program functions for possible native code compilation

is created, and a plurality of program functions from the list of program functions is selected. The selected program functions are precompiled into native program functions.

However, there is simply no disclosure in Kwong of precompiling a class file included in a standard class library into an extended class library file including a machine instruction as required by claim 7. The Examiner cites column 5, lines 8-67 of Kwong as allegedly disclosing this feature of the claim. However, this cited portion of Kwong does not disclose precompiling a class file included in a standard class library into an extended class library file including a machine instruction as required by claim 7.

Accordingly, Applicant respectfully submits that independent claim 7 should be allowable, because the cited reference does not teach or suggest all of the features of the claim. Claims 8, 9, and 16 should also be allowable at least by virtue of their dependence on independent claim 7.

C. Rejection of Claim 1

The Examiner has rejected claim 1 under 35 U.S.C. § 103(a) as being unpatentable over Zee in view of Baentsch.

Claim 1 recites in part:

an extended class library which includes a class file of a machine code obtained by precompiling a class file included in a standard class library; and a Java Virtual Machine (JVM) which executes the class file of the machine code class file or an application file included in the extended class library.

The Examiner alleges that Zee discloses all of the features of claim 1 except for a Java Virtual Machine (JVM), and asserts that it would have been obvious to one of ordinary skill in the art to use the JVM disclosed by Baentsch as a software-only platform running on top of the hardware-based platform of processing systems 30 and 32 disclosed by Zee.¹

However, there is no teaching or suggestion in Zee of an extended class library which includes a class file of a machine code obtained by precompiling a class file included in a standard class library as recited in claim 1. The Examiner cites paragraphs [0060]-[0064] of Zee as allegedly disclosing this feature of the claim. However, this cited portion of Zee merely discloses selecting an Ahead of Time (AOT) compiler which can produce a native component for the data processing system 30 or 32 from a compile information table and driving a corresponding AOT compiler according to an execution condition set in the compile information table to compile a downloaded Java class file into the native component.

Furthermore, in paragraphs [0060]-[0064] of Zee, only the constitution that the AOT compiler compiles the downloaded java class file to a native content and saves the native content is disclosed. However, the claimed invention is not related only to java class files, but to a class file included in a standard class library. Additionally, in the claimed invention an extended class library which includes a class file of a machine code obtained by precompiling a class file included in a standard class library already exists in a java execution device. This feature is simply not taught or suggested by Zee.

¹ Page 11 of the Office Action dated October 19, 2006.

AMENDMENT UNDER 37 C.F.R. § 1.111
U. S. PATENT APPLICATION NO.: 10/621,446

Attorney Docket No.: Q75479

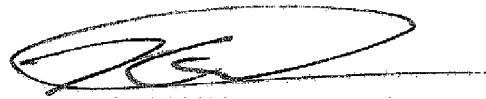
Accordingly, there is simply no disclosure in Zee of an extended class library which includes a class file of a machine code obtained by precompiling a class file included in a standard class library as required by claim 1. Baentsch fails to cure the deficiency of Zee.

Accordingly, Applicant respectfully submits that independent claim 1 should be allowable, because the cited reference does not teach or suggest all of the features of the claim. Claims 2, 3, 14, 15, and 17 should also be allowable at least by virtue of their dependence on independent claim 1.

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,



Mark E. Wallerson
Registration No. 59,043

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE

23373

CUSTOMER NUMBER

Date: January 19, 2007